ADD

```
ADD [ROUNDED] operand1... TO operand2
```

Operand	Possible Structure				Possible Formats								Referencing Permitted	Dynamic Definition
Operand1	C	S	A	N	N	P	I	F	D	T			yes	no
Operand2		S	A	M	N	P	I	F	D	T			yes	yes

```
ADD [ROUNDED] operand1... GIVING operand2
```

Operand	Possible Structure				Possible Formats											Referencing Permitted	Dynamic Definition
Operand1	C	S	A	N		N	P	I	F		D	T				yes	no
Operand2		S	A	M	A	N	P	I	F	B	D	T				yes	yes

Related Statement: COMPUTE

Function

The ADD statement is used to add two or more operands.

Operands

At the time the ADD statement is executed, each operand used in the arithmetic operation must contain a valid value.

For additions involving arrays, see also the section Arithmetic Operations with Arrays.

As for the formats of the operands, see also the section Performance Considerations for Mixed Formats.

Result Field - operand2

TO

If the keyword TO is used, *operand2* will be included in the addition and will contain the result of the addition.

GIVING

If the keyword GIVING is used, *operand2* will be used to store the result only. If GIVING is used and *operand2* is defined with alphanumeric format, the result will be converted to alphanumeric.

If a database field is used as the result field, the addition only results in an update to the internal value that is used within the program. The value of the field in the database is not affected.

ROUNDED

If the keyword ROUNDED is used, the result will be rounded. For rules on rounding, see the section Rules for Arithmetic Assignment.

Example

```
  * EXAMPLE 'ADDEX1': ADD
  ***********************
  DEFINE DATA LOCAL
   1 #A (P2)
   1 #B (P1.1)
   1 #C (P1)
   1 #DATE (D)
   1 #ARRAY1 (P5/1:4,1:4) INIT (2,*) <5>
   1 #ARRAY2 (P5/1:4,1:4) INIT (4,*) <10>
  END-DEFINE
  *
  ADD +5 -2 -1 GIVING #A
  WRITE NOTITLE 'ADD +5 -2 -1 GIVING #A' 15X '=' #A
  *
  ADD .231 3.6 GIVING #B
  WRITE /        'ADD .231 3.6 GIVING #B' 15X '=' #B
  *
  ADD ROUNDED 2.9 3.8 GIVING #C
  WRITE /        'ADD ROUNDED 2.9 3.8 GIVING #C' 8X '=' #C
  *
  MOVE *DATX TO #DATE
  ADD 7 TO #DATE
  WRITE / 'CURRENT DATE:'     *DATX (DF=L)13X
          'CURRENT DATE + 7:' #DATE (DF=L)
  *
  WRITE /        '#ARRAY1 AND #ARRAY2 BEFORE ADDITION'
        /        '=' #ARRAY1 (2,*)
                 '=' #ARRAY2 (4,*)
  ADD #ARRAY1 (2,*) TO #ARRAY2 (4,*)
  WRITE /        '#ARRAY1 AND #ARRAY2 AFTER ADDITION'
        /        '=' #ARRAY1 (2,*)
                 '=' #ARRAY2 (4,*)
  *
  END
```

```
 ADD +5 -2 -1 GIVING #A              #A:   2

 ADD .231 3.6 GIVING #B              #B:  3.8

 ADD ROUNDED 2.9 3.8 GIVING #C       #C:  7

 CURRENT DATE: 1999-01-19            CURRENT DATE + 7: 1999-01-26


 #ARRAY1 AND #ARRAY2 BEFORE ADDITION
 #ARRAY1:     5     5     5     5 #ARRAY2:     10    10    10    10

 #ARRAY1 AND #ARRAY2 AFTER ADDITION
 #ARRAY1:     5     5     5     5 #ARRAY2:     15    15    15    15
```